The ACL RD-TEC Annotation Guideline

A Reference Dataset for the Evaluation of Automatic Term Recognition and Classification

Version 2.6

Anne-Kathrin Schumann and Behrang QasemiZadeh

anne.schumann@mx.uni-saarland.de, me@atmykitchen.info

Language is deceptive and annotation tasks are like deciding the colour of a chameleon.

Abstract

The annotation task: Given a document consisting of a set of sentences, the goal is to mark boundaries of terms in each sentence. Annotated terms are then further classified as instances of semantic categories (concept categories).

Purpose of this document: This document provides readers with basic knowledge that is required to successfully perform the annotation task proposed in this document.[1]

Instructions: Annotators are assumed to read this document carefully before commencing the annotation task. For readers who are familiar with this document, note that a history of changes to this document is provided in Appendix A.

Document organisation: This document is organised in five parts. Section 1 provides basic definitions. After reading this section, an annotator is assumed to understand the meaning of *term* and *concept*.

Section 2 provides information about the annotation procedure. Section 3 outlines termhood criteria intended to help annotators when identifying terms, marking term boundaries and assigning semantic classes. Section 4 provides information about tools that must be used by annotators. Section 5 provides an actual annotation example.

[1]The previous version of this guideline can be found in QasemiZadeh (2014).

1 Basic Definitions

This section provides verbose definitions and background knowledge for the annotators. Section 1.1 describes what is a *term*. Section 1.2 explains the relationship between terms and concepts. Section 1.3 gives a summary and explains these definitions in the context of the annotation task.

1.1 Terms

When we read *specialized texts* (e.g., technical reports, and scientific publications), it is very likely that we encounter items of *specialized vocabulary*, which are called *terms*. Terms are words or phrases that express important information about a domain of knowledge.

Terms have important linguistic functions that are best illustrated by an example: Figure 1 is an extract about rule-based machine translation from the English Wikipedia. This extract contains many specialized terms, for example, "transfer-based machine translation" or "semantic analysis". These terms are central to understanding the text. Now, removing the terms from the Wikipedia extract results in the loss of *textuality*—the text becomes dysfunctional. Without terms, as

> The rule-based machine translation paradigm includes transfer-based machine translation, interlingual machine translation and dictionary-based machine translation paradigms. This type of translation is used mostly in the creation of dictionaries and grammar programs. Unlike other methods, RBMT involves more information about the linguistics of the source and target languages, using the morphological and syntactic rules and semantic analysis of both languages. The basic approach involves linking the structure of the input sentence with the structure of the output sentence using a parser and an analyzer for the source language, a generator for the target language, and a transfer lexicon for the actual translation.

Figure 1: Example extract from Wikipedia: Rule-based machine translation

> The [redacted] paradigm includes [redacted] [redacted] and [redacted] paradigms. This type of [redacted] is used mostly in the creation of [redacted] and [redacted]. Unlike other methods, [redacted] involves more information about the [redacted] of the [redacted], using the [redacted] and [redacted] of both [redacted]. The basic approach involves linking the structure of the [redacted] with the structure of the [redacted] using a [redacted] and an [redacted] for the [redacted], a [redacted] for the [redacted], and a [redacted] for the actual [redacted].

Figure 2: Example extract from Wikipedia: The same text snippet in Figure 1 but terms are removed.

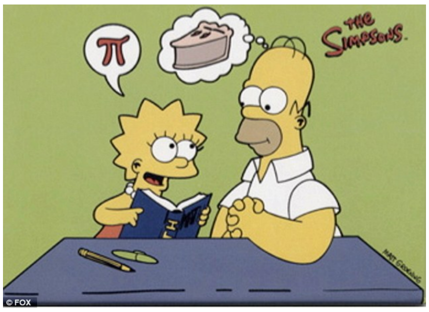

Figure 3: Lexical gap between expert and layman.

shown in Figure 2, this text loses not only its informativeness but also important properties such as coherence and cohesion. It is hardly recognizable as a text! It is clear from this example that terms are an important means of communicating specialized knowledge.

But what makes terms (i.e., specialized vocabulary) "special" in comparison to general language words? Most importantly, terms are used in specialized professional contexts rather than in everyday life. A well-known example is the jargon used by medical doctors: it is often hard to understand for their patients, even if they speak the same language (e.g., "English"). Put simply, knowledge of English does not necessarily result in knowledge of medical terms. This is perhaps because medical knowledge is in most cases not necessary for successful communication in our everyday lives. There is a *lexical gap* between experts and laymen and this gap is one of the reasons why specialized texts are sometimes hard to read and understand (see Figure 3).

1.2 Terms and Concepts

The relationship between terms and specialized knowledge can be described by stating that a term is a word or phrase referring to a concept in a specialized subject field. According to ISO 1087-1(2000), a term is 'a verbal designation of a general concept in a specific subject field (ISO, 2000).'

So what exactly are concepts and what is the relation between terms and concepts? *Concepts* can be understood as complex units of knowledge[2]. They are semantic units, abstract mental entities that are dealt with by specialized texts, for example, *MACHINE TRANSLATION* in the domain of computational linguistics, *COMBUSTION ENGINE* in the domain of mechanical engineering, or *IMMUNE SYSTEM* in the domain of medicine. Terms, then, are *linguistic labels* or *names*

[2] Others still have found different names for concepts, e.g. "units of understanding" (Temmerman, 2000) or units of thought (Wüster, 1979).

for concepts that are used to communicate about concepts (L'Homme, 2014). They allow us to handle concepts in a linguistically efficient way, for example, by avoiding the repetition of lengthy explanations and definitions.

How does this work? Even if we deal with a very complicated concept, we can still assign a relatively simple name (a term) to it and use this name when we want to explain our thoughts to others. For example, saying "rule-based machine translation" is much shorter and easier than saying "the automatic transfer of text from one source language to a target language using linguistic rules ...". Accordingly, terms facilitate text production by allowing us to pack conceptual knowledge into flexible linguistic units. Last but not least, terms can be used as a base for the derivation of new lexemes (e.g., "domain-specific", derived from "domain").

An important result of the distinction between *concepts* and *names for concepts* (i.e., terms) is that different names can be used to talk about the same concept. However, even if this occurs, we can still group together all the terms that refer to the same concept, provided that we have sufficient expertise about the domain in question. Given this perspective, a concept and its corresponding terms can be seen as a tuple $(\{T\}, c)$, where $\{T\}$ is a set of terms that appear in a text whenever this text deals with the concept c. $\{T\}$ then comprises all the different variants and linguistic surface forms $\{t_1, ..., t_n\}$ that refer to the same concept c. For example, both "rule-based machine translation" and "RBMT" refer to the same concept, namely *RULE-BASED MACHINE TRANSLATION*.

Also, it is important to note that tuples $(\{T\}, c)$ are valid only in a certain specialized domain D. For example, "tree" is a term in computer science, but it is also a general language word. Or consider the term "solution" whose meaning in chemistry is totally different from its meaning in mathematics. These delicate relationships between terms and concepts are not annotated in this task. However, annotators must bear in mind these relationships when commencing annotations.

1.3 Wrap-Up

To wrap up this discussion about the terminological foundation of the task proposed here, we restate that concepts are units of specialized knowledge in a given subject field. And terms are the linguistic expressions that people use to talk about them.

2 An Overall View of the Annotation Workflow

The term annotation task proposed in this document consists of the two subprocedures explained below:

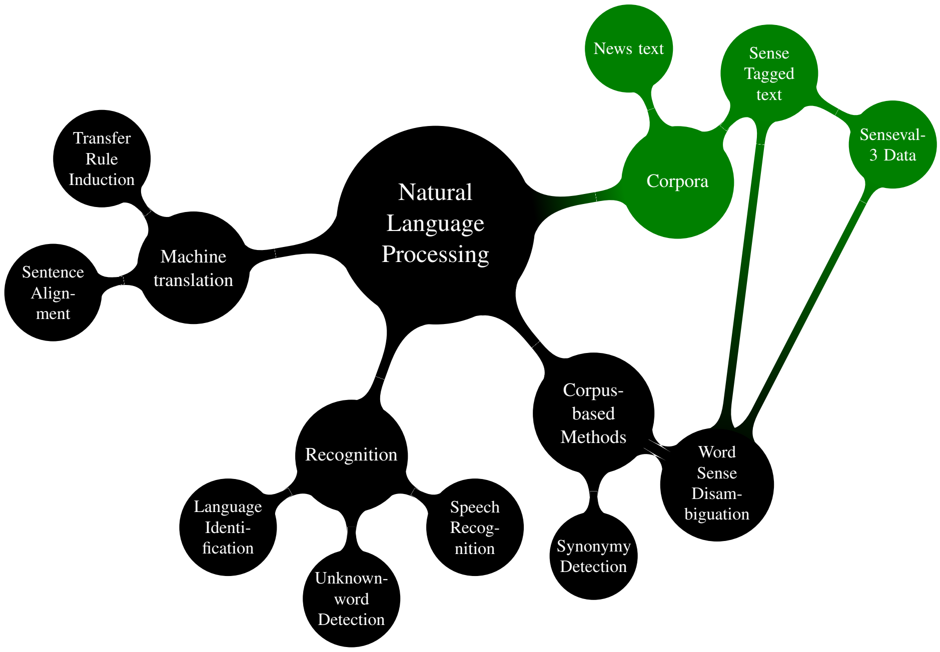

Figure 4: Example of a mind-map: black nodes represent technologies, while green nodes show other terms.

(a) **Identification of concepts and terms:** In this process, annotators identify concepts that are relevant to the given domain of knowledge. They mark the border of the terms that correspond to these concepts in a set of texts pertaining to that domain. The targeted domain of knowledge is *computational linguistics*. When reading a text that needs to be annotated, annotators must ask themselves whether any given lexical unit *activates* knowledge (concepts) about computational linguistics in their minds. An easy way to do this is to imagine a mind-map of the topics in computational linguistics. Would you like to see a given concept (thus term) in this map (see Figure 4)? If the answer is yes, then the lexical unit which activates this concept should be annotated as a term.

(b) **Assignment of terms and concepts to semantic classes**: Once annotators identify a term in a text snippet, they elaborate its semantics by means of assigning the identified term to one of the predefined semantic categories that are listed in Table 1.

In this round of annotations, the aim is to mark all mentions of terms in the annotated data (i.e., a set of selected abstracts). No further grouping of terms (e.g., into sets of *variants*) is pursued at this stage. This is done in order to keep the annotation task as simple as possible. Further analyses including the annotation of term variants will be carried out in a separate annotation step.

In the following sections, annotators are provided with practical guidelines to perform the task. Section 2.1 gives an overview of the text-data they deal with. Section 2.2 delineates the procedure employed to commence the annotation.

2.1 Data

In this project, the data is drawn from the ACL RD-TEC (QasemiZadeh and Handschuh, 2014). This corpus comprises preprocessed scientific publications from the domain of computational linguistics as published in the ACL Anthology Reference Corpus (ACL ARC) (see Bird et al., 2008).

Annotators are provided with simple XML files. These text files are abstracts from the publications in the ACL ARC. Annotators must not change the logical structure of these files such as the naming of XML markups. Listing 1 shows an example of an XML file that will be provided to annotators. Annotators are expected to append markups to these files.

Markups inserted by annotators are in the form of

```
...<NUMBER>TERM</NUMBER> ....
```

In this structure, `<NUMBER>` and `</NUMBER>` show the boundary of a term `TERM`. `NUMBER`, in turn, is an integer that is determined by the semantic category of the annotated term. Further details about how to identify the boundaries of terms and how to perform the annotations are provided in Section 3. An actual annotation process is exemplified in Section 5 in which an annotated XML file and the set of questions that may arise during the process are listed.

2.1.1 Annotator Roles

In this project and during the annotation task, we will distinguish between two annotator roles:

- primary annotators, who will do the actual markup of term candidates/identification of term boundaries in the corpus texts;
- the lead annotator(s) who will resolve problems due to conflicting annotation. The lead annotators decide whether changes in the guidelines are necessary. Also, the lead annotators are in charge of versioning of the data, training primary annotators, and enforcing consistency in the annotations as described in Section 2.2.1.

Listing 1: Example of an abstract file given to an annotator

```
<?xml version="1.0" encoding="UTF-8"?>
<Paper uid="C02-1042">
  <Title>Using Knowledge to Facilitate Factoid Answer Pinpointing</Title>
  <Section position="1" start_page="0" end_page="0" type="abstr">
    <SectionTitle>Abstract</SectionTitle>
    <Paragraph position="0">In order to answer factoid questions, the
        ↪ Webclopedia QA system employs a range of knowledge resources.
        ↪ These include a QA Typology with answer patterns, WordNet,
        ↪ information about typical numerical answer ranges, and
        ↪ semantic relations identified by a robust parser, to filter
        ↪ out likely-looking but wrong candidate answers. This paper
        ↪ describes the knowledge resources and their impact on system
        ↪ performance.</Paragraph>
  </Section>
</Paper>
```

2.2 Preparing Annotators for the Task

To ensure quality and consistency in annotations, the annotation task is carried out in two steps, namely:

- an *annotation training phase* serving to assure mutual understanding of these guidelines.
- the actual annotation phase.

During the annotation training phase, each annotator will work on a set of 10 abstracts individually. After the completion of this test set, meetings will be held to discuss results and possible conflicting points in these guidelines. This procedure may result, if necessary, in amending the guidelines. The annotation of this small set of 10 abstracts can be repeated to ensure annotators are ready for the actual annotation phase.

During the actual annotation phase, annotations will only be revised by the lead annotators.

2.2.1 Resolving Conflicts

Conflicts can occur in two cases, namely when:

- *mismatch of term/concept identification*: at least two annotators mark different, disjunct strings of word tokens;
- *mismatch of term boundary definition*: when at least two annotators mark different, but overlapping strings of word tokens.

In the first case, different annotators exhibit a mismatch in what they consider to be relevant computational linguistics concepts or valid terms (i.e., names for computational linguistics concepts). These cases will not be normalized by the lead annotator. Instead, information about how many annotators annotated a given string will be kept in the annotation metadata. It is important to note that conflicts of this type are expected considering the complexity of the annotation task.

Annotation conflicts of the second type, however, are the mismatches concerning the actual structure of a given term—that is, which linguistic constituents are indispensable parts of this term. In these cases, the lead annotators will ensure consistent term annotation throughout the whole project and in agreement with guidelines given in Section 3.

3 Termhood Criteria

When considering the status of a candidate term (i.e., deciding about the boundaries of a term and its semantics), annotation decisions should be based on three types of *termhood* criteria, namely *semantic*, *linguistic*, and *formal criteria*. Moreover, annotators are actively encouraged to collect more information about linguistic units under consideration, for example, using the Sketch Engine's on-line installation of the ACL ARC[3]. On- and offline search in other resources is, of course, also allowed.

3.1 Semantic criteria

In this project, terms are annotated and classified to certain semantic classes. Table 1 gives an overview of semantic classes together with examples, which are explained below:

- **Technology, System, and Method** terms refer to methods, processes, and approaches that are employed to solve practical tasks. In computational linguistics, "machine translation", "information extraction", "word sense disambiguation", ... are examples of technology terms. Often, the head noun is derived from a verb that describes a practical activity (e.g. "to analyze" and its nominalisation "semantic analysis") or process (e.g. "to propagate" and its nominalisation "(constraint) propagation", found in ACL RD-TEC). In running text, such terms are sometimes accompanied by generic nouns such as "paradigm", "approach" or "method" (e.g. "rule-based machine

[3] Accessible from https://the.sketchengine.co.uk/bonito/run.cgi/corp_info?corpname=preloaded/aclarc_1.

#	Semantic class	Examples
1	Technology, System, and Method	rule-based machine translation, transfer-based machine translation, interlingual machine translation, dictionary-based machine translation, RBMT, semantic analysis, speech synthesis, speech recognition, …
2	Tool or Library	Stanford Core NLP, NLTK, OpenNLP library, Sphinx, CMUnicator, …
3	Language resource	dictionary, grammar, transfer lexicon, syntactic rule, lexical selection rule, core grammar, …
4	Language Resource Product	WordNet, Brown Corpus, SentiWordNet, Reuters-21578 , …
5	Models	language model, translation models, word-based language model, error model, maximum entropy model, n-gram models, …
6	Measures and Measurements	BLEU, NIST, Precision, Recall, F-Score, machine translation tests, MT evaluation, …
0	Other Nominals (theories, formalism, …)	target language, language, input sentence, output sentence, source language, orthographical variation, input, ambiguity, domain, lexical selection, Japanese, phrase, sentence, …

Table 1: Semantic classes of terms with examples

translation paradigm" in the Wikipedia extract). This category of terms is marked by number "1" in the annotations.

- **Tool or library** terms refer to an actual implemented technology. Terms that belong to this class can be understood as instances of technology terms.[4] In the domain of computational linguistics, *tools* are often computer programmes used to carry out the actual analyses. For example, while "part of speech tagger" is a technology term, the terms "TreeTagger" and "Stanford PoS Tagger" belong to the category of *tools and libraries*. This category of terms is marked by number "2" in the annotations.

- **Language resources** are mainly components of natural language processing (NLP) systems that contain linguistic knowledge, for example, lexical databases, corpora, and so on. However, this category of terms does not specify a particular language resource product (see below). This category of terms is marked by number "3" in the annotations.

- **Language resources product** terms refer to actual language resources. For example, "Princeton WordNet" is a lexical database which can be obtained and used in a project. These actual resources are marked by number "4" in the annotations.

[4] In other words, the *tool or library class* is a sub-category of the technology class.

- **Models** refer to method-specific knowledge resources. For example, "language model" is usually a database of the probability distribution of a sequence of words that is employed by a method to perform a task (e.g., part-of-speech tagging). Similarly, "phrase tables" and "translation models" are knowledge resources employed by particular types of machine translation technologies. It is worthwhile mentioning that *model* terms sometimes act as the referent to technologies. In these cases, terms must be annotated by their meaning in context (i.e., as technology). This category of terms are marked by number "5" in the annotations.

- **Measures and Measurements** comprise terms referring to measures (e.g. "BLEU", "f-score") or more abstract measurements (e.g. "accuracy" or "translation quality"). These terms are marked by number "6" in the annotations.

- Finally, the **other** are any other abstract concepts in the field of computational linguistics that cannot be fitted into any of the above listed classes. Therefore, this class of terms encompasses a large variety of terms such as linguistic entities, scientific disciplines and so on. Terms with a very specific meaning in a highly specialised context can also belong to the *other* category. For example, in the context of machine translation, *language pairs* such as "Czech-English" fall under the *other* class. These terms are marked by number "0" in the annotations.

3.2 Linguistic criteria

A range of linguistic rule-of-thumb criteria can be used to identify terms such as:

- **Etymology**: In European languages, many terms are of Greek or Latin origin. For many of the terms in Table 1 this is actually the case.

- **Comprehensibility**: As pointed out before, even if annotators are competent speakers of English, this does not mean that they understand all English terms irrespective of the domain. Specialized terminology often is incomprehensible for a layman. If, during annotation, annotators feel that a given unit is strongly bound to specialized knowledge on computational linguistics, then they might have found a term.

- **Distribution**: Unsurprisingly, terms are not distributed evenly across texts: they are more frequent in specialized texts of a given domain and much less frequent in all other texts. This property has been exploited in many statistical term extraction algorithms. So if, during annotation, annotators find that a given unit is rather rare or “weird”, they may have found another term.

- **Morphosyntactic patterns**: Most terms are noun phrases. In this project, we also mainly annotate *noun phrases*.

3.3 Formal criteria

There are also some formal criteria for annotating terms:

- **Determiners**: Do not annotate determines or pronouns of any kind as part of a term. For example, in the string ”a machine translation method”, the term is ”machine translation method”.

- **Abbreviations**: Abbreviations can be terms if they designate specialized concepts, for example, RBMT or TF-IDF. These abbreviations are shorthands for expressing terms (e.g., RBMT for “rule-based machine translation”).

- **Term-abbreviation sequence:** In the case that a term is followed by its abbreviation, the whole sequence is annotated as one term. For instance, given the text “machine translation (MT)”, the whole sequence is annotated as one term (instead of annotating “machine translation” and “MT” as separate terms).

- **Terms broken by abbreviations**: In a number of occasions, an abbreviation of a general term is inserted into a term of more specific meaning. For example, in “machine translation (MT) evaluation”, “MT” is the abbreviation of the term ”“machine translation” which is inserted in the more specific term ”machine translation evaluation”. In these circumstances, the whole sequence is annotated as one term. The semantic category of the term is decided by the semantics of the more specific term, namely “machine translation evaluation”.

- **Proper nouns**: Proper nouns (names) should be annotated only if the related concept belongs to one of the categories listed in Table 1 such as the *Tool or Library* (e.g., TreeTagger, ABNER, . . .), the *Language Resource Product* (e.g., EuroWordNet, WordNet, . . .) or the *Other* categories (e.g., conferences such as LREC or associations such as ELRA, ACL, . . .). Other kinds of names (e.g., people or place names) should not be annotated.

- **Generic nouns**: As explained before, terms can be accompanied by generic nouns, e.g., the word “approach” in “sequential labelling approach”. Although it can be argued that “approach” is a generic noun and the term is “sequential labelling”, for a number of considerations[5] annotators are asked to mark “sequential labelling approach” as a term. We call this principle the **maximal length annotation** principle.

- **Adjectival modifiers**: In many cases, terms are modified by adjectives, for example, in the strings “systematic pattern” and “statistical MT”. How to decide whether the adjective is part of the term or not? Try by checking whether removing the adjective changes the meaning of the term or not. In the case of our examples, we see that saying “pattern” instead of “systematic pattern” does not change much since patterns are by default something that is systematic. “MT”, however, is more general than “statistical MT” since there are also many other approaches to machine translation. In the first case, annotate the term *without* the redundant modifier (that is, “pattern” instead of “systematic pattern”). In the latter case, annotate the complete span of the noun phrase (“statistical MT” instead of “MT”).

- **Conjunctions and prepositions**: Complex term phrases can contain conjunctions and prepositions, for example, “TREC 2003 and TREC 2004 QA tracks”, or “automatic evaluation of machine translation and document summarization”. In these cases, a set of rules applies. The main goal of providing these rules is to facilitate future automatic manipulations of manually provided annotations.

  - For conjunctions, if the noun phrases linked by them are *ellipses*, the whole span should be annotated as one. For example, in “supervised and unsupervised methods”, where we can also read “supervised methods and unsupervised methods”, “supervised and unsupervised methods” is annotated as *one term*. Otherwise, split the string at the conjunction and annotate the conjuncts separately.
  - Complex phrases containing *prepositions* can normally be split at those points where the prepositions are placed. Thus, for the text snippet ”automatic evaluation of machine translation and document summarization”, “automatic evaluation”, “machine translation”, and “document summarization” are annotated as separate terms.

    However, it is important to note that sometimes prepositions are a part of the term. Familiar examples are terms such as ”text to speech”, ”part

[5]For example, ease of automatic manipulation of annotations in later stages.

of speech tagging”, ... Similarly, in relaxed paraphrases of normally denser terms (e.g., in “method for recognizing systematic patterns” instead of “pattern recognition method”), the whole sequence must be annotated as one term.

- In general, we aim at a *greedy annotation*. Some examples of what this means are given below:

  Example:

  * As explained earlier, given the string ”statistical machine translation method” which represents a single concept, the term is “statistical machine translation method”.
  * Given the string “supervised training data” in:

    > This paper presents a maximum entropy word alignment algorithm for Arabic-English based on supervised training data.

    annotate what is intended, (i.e., “supervised training data” as one term referring to ”manually annotated training data”). In other words, in this example *do not* annotate ”supervised training” and ”data” as two separate terms.
  * Given a nested term consisting of several concepts/terms (e.g., “maximum entropy word alignment algorithm”), annotate it as one term.
  * Given ad-hoc formations, (e.g., “suffix array-based data structure”), annotate them as a whole.

- **Incorrect spelling**: Annotators must be reminded that the text they are provided with can contain noise. Such being the case, they are requested to correct miss-spelled text (e.g., “wordsense disanbiguation” must be edited to “word sense disambiguation”). Similarly, the titles of the abstracts may be incorrect due to the errors from the segmentation and OCR processes. Annotators are requested to edit these fields, too.

Last but not least, in general and as suggested earlier, it is important for annotators to consider the meaning of candidate terms in context. If a candidate term plays a role in conveying a specialised meaning (i.e., a concept of a particular category), then it must be annotated. Annotators must bear in mind that deciding “what a term is” is partly dependent on their conceptualisation of the domain and their understanding of the written text. To ease the decision-making process, apart from criteria set out above and the semantic categories that are listed in Table 1, we provide a number of more fine-grained distinctions about what to annotate and what not to, listed in Table 2.

Category	Example	**Term?**
Domain high-level terms	algorithm, phrase, ...	YES
General scientific terms	experimental result, ...	NO
Contextual synonyms	e.g., "alignment" (when it is used instead of "word alignment"); "non-contiguous phrases" (to emphasise particular type of "phrases")	YES
paraphrases of more compact terms	e.g., "method for recognizing systematic patterns" instead of "pattern recognition method"	YES
Co-referencing items	e.g., "this analysis" or "this topic" vs. "semantic analysis" where "topic" and "analysis" refer to "semantic analysis" mentioned earlier in the text	NO

Table 2: Fine-grained semantic categories for annotation

4 Preparing Final Annotation Files

To perform the annotation task and inserting markups, we suggest annotators use `Notepad++`[6] (for Windows users) and `GNU Emacs` for Linux and Mac users.

Once annotators finish their annotations, they are supposed to convert and verify their raw annotations using the "annotation conversion tool".[7] This tool is designed to:

a) make sure that annotations are consistent. For example, the opening and ending tags are correct, the employed markups are consistent with those suggested in this guideline, etc;

b) convert the number-based annotation files to the valid XML annotation schema designed by the lead annotators.

In case of errors in an annotation file, the system suggests an offset (i.e., line number/column number) in which an error has taken place.

After the conversion, the generated annotation files in addition to the guidelines employed by the annotators must be handed over to the lead annotators for further inspection.

[6]Download from https://notepad-plus-plus.org/

[7]Available on-line from https://bitbucket.org/aclrdtee/acl-rd-tee/downloads/complied_annotation_convert_ver1.zip.

Listing 2: Example of an abstract file annotated manually

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Paper uid="C02-1042">
  <Title>Using Knowledge to Facilitate Factoid Answer Pinpointing</Title>
  <Section position="1" start_page="0" end_page="0" type="abstr">
    <SectionTitle>Abstract</SectionTitle>
    <Paragraph position="0">In order to answer <0>factoid questions</0>,
       ↪  the <2>Webclopedia QA system</2> employs a range of <3>
       ↪ knowledge resources</3>. These include a <3>QA Typology</3>
       ↪ with <0>answer patterns</0>, <4>WordNet</4>, information about
       ↪  typical <0>numerical answer</0> ranges, and <0>semantic
       ↪ relations</0> identified by a robust <1>parser</1>, to filter
       ↪ out likely-looking but wrong <0>candidate answers</0>. This
       ↪ paper describes the <3>knowledge resources</3> and their
       ↪ impact on <0>system performance</0>.
    </Paragraph>
  </Section>
</Paper>
```

5 Exemplified Annotation Process

Listing 2 provides an example of the annotation process carried over the abstract presented in listing 1.

Some questions that may arise during the annotations are:

- Is “Webclopedia QA system” the term or “Webclopedia QA”?
- Is “QA” a term to be annotated in this context or “QA system”, or none?

The answers to these questions are given in Section 3.3.

Once the annotation is performed, the XML version is generated automatically using the provided tool (e.g., as shown in Listing 3).

References

Bird, S., Dale, R., Dorr, B., Gibson, B., Joseph, M., Kan, M.-Y., Lee, D., Powley, B., Radev, D., and Tan, Y. F. (2008). The acl anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of LREC’08*, Marrakech, Morocco. ELRA. 6

ISO (2000). *ISO 1087-1:2000: Terminology work – Vocabulary – Part 1: Theory and application*. ISO, Geneva. 3

Listing 3: Example of an XML annotation file generated by the conversion tool

```
<?xml version="1.0" encoding="UTF-8"?>
<Paper uid="C02-1042">
  <Title>Using Knowledge to Facilitate Factoid Answer Pinpointing</Title>
  <Section position="1" start_page="0" end_page="0" type="abstr">
    <SectionTitle>Abstract</SectionTitle>
    <Paragraph position="0">In order to answer <term class="other">
       ↪ factoid questions</term>, the <term class="tool">Webclopedia
       ↪ QA system</term> employs a range of <term class="lr">knowledge
       ↪  resources</term>. These include a <term class="lr">QA
       ↪ Typology</term> with <term class="other">answer patterns</term
       ↪ >, <term class="lrp">WordNet</term>, information about typical
       ↪  <term class="other">numerical answer</term> ranges, and <term
       ↪  class="other">semantic relations</term> identified by a
       ↪ robust <term class="tech">parser</term>, to filter out likely-
       ↪ looking but wrong <term class="other">candidate answers</term>
       ↪ . This paper describes the <term class="lr">knowledge
       ↪ resources</term> and their impact on <term class="other">
       ↪ system performance</term>.
    </Paragraph>
  </Section>
</Paper>
```


L'Homme, M.-C. (2014). Terminologies and taxonomies. In Taylor, J. R., editor, *The Oxford Handbook of the Word*. Oxford University Press. 4

QasemiZadeh, B. (2014). *The ACL RD-TEC: A Reference Dataset for the Evaluation of Automatic Term Recognition and Classification in Computational Linguistics: Annotation Guideline*, version 1 edition. retrieved from http://atmykitchen.info/datasets/acl_rd_tec/annotation_guideline/annotation-guideline.pdf. 1

QasemiZadeh, B. and Handschuh, S. (2014). The acl rd-tec: A dataset for benchmarking terminology extraction and classification in computational linguistics. In *Proceedings of the Computerm'14*. 6

Temmerman, R. (2000). *Towards new ways of terminology description : the sociocognitive-approach*. John Benjamins, Amsterdam. 3

Wüster, E. (1979). *Einführung in die allgemeine Terminologielehre und terminologische Lexikographie*, volume 1: Textteil of *Schriftenreihe der Technischen Universität Wien; 8*. Springer, Wien. 3

Appendix A Summary of Changes

Ver	Date	Added to repository
2.0	15.07.2015	Document Created by Anne.
2.1	28.08.2015	Anne applied the following changes: • Table 2 added; • Linguistic examples added to Table 1; • Section 3.3 is edited.
2.2	01.09.2015	Behrang changed the guidelines: • Document is restructured. This section is moved to appendices and linguistic examples are moved to Section 3.3. • Section 4 is added to the document. • Section 3.3 is edited. • Section 5 is edited; new listing is added.
2.3	15.09.2015	Anne changed the guidelines: • Section 3.1 is edited. • Section 3.2 is edited. • section 3.3 is edited.
2.4	18.09.2015	Anne changed the guidelines: • Table 2 is edited.

2.5	20.09.2015	Behrang edited the guidelines: • Overall editing of this document. • Small changes are made in Section 1.2. • Section 2 is restructured. • Semantic classes in Section 3.1 are edited and new categories are added. • Section 3.3 (i.e., formal criteria) is edited.
2.6	13.10.2015	Anne and Behrang changed the guidelines: • Two new semantic classes (*model*, and *measure and measurements*) are added.